PRA

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/468,051	12/20/1999	THOMAS D. HARTNETT	RA-5271	3159

7590 06/19/2003

UNISYS CORPROATION
ATTN BETH L MCMAHON
M S 4773
P O BOX 64942
ST. PAUL, MN 551640942

EXAMINER
WOOD, WILLIAM H

ART UNIT	PAPER NUMBER
2124	12

DATE MAILED: 06/19/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/468,051	Applicant(s) HARTNETT ET AL.	
	Examiner William H. Wood	Art Unit 2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 April 2003* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-46* is/are pending in the application.

4a) Of the above claim(s) *8-20* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7 and 21-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Claims 1-7 and 21-46 have been examined.

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 30 April 2003 has been entered.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 22, 23, 28, 29, 33, 34, 38 and 45 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. New matter is: *instruction address generate logic* (claims 22, 23, 28, 29, 33 and 34); *circuit to allow retrieval of an instruction from either the memory or from the queue* (claim 38); and *indication that one or more operations are occurring within execution stages and in*

response to the indication, allowing instructions to advance within the fetch stages (claim 45).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 21, 27, 32, 39, 40 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhamidipati et al. (USPN 6,112,295) in view of Hayes, John P., "Computer Architecture and Organization".

In regard to claim 1, Bhamidipati disclosed the limitations:

- *For use in an instruction processor that executes instructions included in a predetermined instruction set* (column 1, lines 51-53), *comprising:*
  - *a pipeline execution circuit to process a first predetermined number of instructions simultaneously* (Figure 1, element 106), *each of said first predetermined number of instructions being in a respectively different stage of execution within said pipeline execution circuit* (Figure 2); *and*
  - *a pipeline fetch circuit coupled to provide each of the first predetermined number of instructions to said pipeline execution circuit* (Figure 1, element 106), *the pipeline fetch circuit to retain a second predetermined number of instructions simultaneously* (Figure 3, elements 300, 302, 304 and 308),

each of said second predetermined number of instructions being in a respectively different stage of processing within said pipeline fetch circuit (Figure 2), *an instruction being capable of advancing to a next stage of execution within said pipeline fetch circuit independently of the times at which instructions advance to a next stage of execution within said pipeline execution circuit* (Figure 3, element 306)

Bhamidipati did explicitly state a synchronous pipeline. However, Bhamidipati indicated that it was known at the time of invention to process instructions in a synchronous pipeline (column 2, lines 43-55; and Figures 1-2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's decoupling queue system in a synchronous pipeline as found in Bhamidipati's own teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a pipeline as close to conventional as possible (easier support in the future). Furthermore, Bhamidipati's teachings suggest this is the desired implementation by discussing this type of pipeline as if it is the type being modified by Bhamidipati's improvement (column 2, lines 43-55; Figures 1-2) and by the great lengths Bhamidipati takes to keep the improvements within a single clock cycle (column 1, lines 50-61).

Bhamidipati did not explicitly state the fetch circuit to transfer each instruction processed by the fetch circuit directly from one of the fetch stages to one of the execution stages. Hayes demonstrated that it was known at the time of invention to directly couple a fetch

stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

In regard to claim 21, Bhamidipati disclosed the limitations:

- *For use in an instruction processor* (column 1, lines 51-53), *comprising:*
  - *an execution circuit to provide a first predetermined number of execution stages* (Figure 1, element 106), *each being capable of performing a respective processing operation on a respective instruction* (execution stages inherently process instructions); *and*
  - *a fetch circuit coupled to the execution circuit to provide a second predetermined number of fetch stages* (Figure 3, elements 300, 302, 304 and 308), *each fetch stage being capable of performing a respective pre-execution operation on a respective instruction* (elements of Figure 3 show operations that are prior to execution), *ones of the instructions processed within the fetch stages being capable of advancing to different*

available fetch stages independently of whether instructions are advancing within the execution stages (Figure 3, element 306).

Bhamidipati did explicitly state a synchronous pipeline. However, Bhamidipati indicated that it was known at the time of invention to process instructions in a synchronous pipeline (column 2, lines 43-55; and Figures 1-2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's decoupling queue system in a synchronous pipeline as found in Bhamidipati's own teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a pipeline as close to conventional as possible (easier support in the future). Furthermore, Bhamidipati's teachings suggest this is the desired implementation by discussing this type of pipeline as if it is the type being modified by Bhamidipati's improvement (column 2, lines 43-55; Figures 1-2) and by the great lengths Bhamidipati takes to keep the improvements within a single clock cycle (column 1, lines 50-61).

Bhamidipati did not explicitly state *the fetch circuit to transfer each instruction processed by the fetch circuit directly from one of the fetch stages to one of the execution stages.* Hayes demonstrated that it was known at the time of invention to directly couple a fetch stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been

obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

In regard to claim 27, Bhamidipati disclosed the limitations:

- *A instruction pipeline circuit for processing instructions within a data processing system* (column 1, lines 51-53), *comprising:*
  - *a first predetermined number of fetch stages to simultaneously process at least a first predetermined number of instructions* (Figure 3, elements 300, 302, 304 and 308);
  - *a second predetermined number of execution stages to simultaneous process a second predetermined number of instructions* (Figure 1, element 106); *and*
  - *wherein at least one of the fetch stages is capable of providing an instruction to a different one of the fetch stages that is ready to receive an instruction irrespective of movement of instructions between the execution stages* (Figure 3, element 306).

Bhamidipati did explicitly state a synchronous pipeline. However, Bhamidipati indicated that it was known at the time of invention to process instructions in a synchronous pipeline (column 2, lines 43-55; and Figures 1-2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's decoupling

queue system in a synchronous pipeline as found in Bhamidipati's own teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a pipeline as close to conventional as possible (easier support in the future). Furthermore, Bhamidipati's teachings suggest this is the desired implementation by discussing this type of pipeline as if it is the type being modified by Bhamidipati's improvement (column 2, lines 43-55; Figures 1-2) and by the great lengths Bhamidipati takes to keep the improvements within a single clock cycle (column 1, lines 50-61).

Bhamidipati did not explicitly state *the fetch circuit to transfer each instruction processed by the fetch circuit directly from one of the fetch stages to one of the execution stages.* Hayes demonstrated that it was known at the time of invention to directly couple a fetch stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

In regard to claim 32, Bhamidipati disclosed the limitations:

- *A instruction pipeline to execute instructions* (column 1, lines 51-53), *comprising:*
  - *an execution circuit having a first predetermined number of execution stages to execute a first predetermined number of instructions substantially simultaneously* (Figure 1, element 106); *and*
  - *a fetch circuit having a second predetermined number of fetch stages to perform pre-execution operations on at least a second predetermined number of instructions substantially simultaneously* (Figure 3, elements 300, 302, 304 and 308), *at least one of the at least second predetermined number of instructions being capable of advancing between different ones of the fetch stages regardless of whether an instruction is being transferred by the fetch circuit to the execution circuit* (Figure 3, element 306).

Bhamidipati did explicitly state a synchronous pipeline. However, Bhamidipati indicated that it was known at the time of invention to process instructions in a synchronous pipeline (column 2, lines 43-55; and Figures 1-2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's decoupling queue system in a synchronous pipeline as found in Bhamidipati's own teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a pipeline as close to conventional as possible (easier support in the future). Furthermore, Bhamidipati's teachings suggest this is the desired implementation by discussing this type of pipeline as if it is the type being modified by

Bhamidipati's improvement (column 2, lines 43-55; Figures 1-2) and by the great lengths Bhamidipati takes to keep the improvements within a single clock cycle (column 1, lines 50-61).

Bhamidipati did not explicitly state *the fetch circuit to transfer each instruction processed by the fetch circuit directly from one of the fetch stages to one of the execution stages.* Hayes demonstrated that it was known at the time of invention to directly couple a fetch stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

In regard to claim 39, Bhamidipati and Hayes disclosed the limitation *wherein the fetch circuit includes a circuit that allows instructions to advance within the second predetermined number of fetch stages if one of the execution stages is performing a predetermined function* (Bhamidipati: Figure 3, element 306).

In regard to claim 40, Bhamidipati disclosed the limitations:

- *A method of processing instructions within a pipeline of an instruction processor* (column 1, lines 51-53), *comprising:*
  - *a.) performing pre-execution operations on a first predetermined number of instructions substantially simultaneously within a first predetermined number of fetch stages of the pipeline* (Figure 3, elements 300, 302, 304 and 308);
  - *b.) executing a second predetermined number of instructions substantially simultaneously within a second predetermined number of execution stages of the pipeline (*Figure 1, element 106*); and*
  - *c.) allowing one or more of the first predetermined number of instructions to advance between ones of the fetch stages independently of whether any of the second predetermined number of instructions are advancing between ones of the execution stages* (*Figure 3, element 306*).

Bhamidipati did explicitly state a synchronous pipeline. However, Bhamidipati indicated that it was known at the time of invention to process instructions in a synchronous pipeline (column 2, lines 43-55; and Figures 1-2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's decoupling queue system in a synchronous pipeline as found in Bhamidipati's own teachings. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a pipeline as close to conventional as possible (easier support in the future). Furthermore, Bhamidipati's teachings suggest this is the desired implementation by discussing this type of pipeline as if it is the type being modified by

Bhamidipati's improvement (column 2, lines 43-55; Figures 1-2) and by the great lengths Bhamidipati takes to keep the improvements within a single clock cycle (column 1, lines 50-61).

Bhamidipati did not explicitly state *wherein each of the second predetermined number of instructions were received directly from one of the fetch stages.* Hayes demonstrated that it was known at the time of invention to directly couple a fetch stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

In regard to claim 46, Bhamidipati disclosed the limitations:

- *A pipeline circuit for use in an instruction processor* (column 1, lines 51-53), *comprising:*
  - *instruction fetch means for performing pre-execution operations on a first predetermined number of instructions substantially simultaneously within a first predetermined number of fetch stages* (Figure 3, elements 300, 302, 304 and 308);

- *instruction execution means for executing a second predetermined number of instructions substantially simultaneously within a second predetermined number of execution stages* (Figure 1, element 106); *and*
- *wherein the instruction fetch means includes means for allowing at least one of the first predetermined number of instructions to advance within the fetch stages irrespective of whether instructions are advancing within the execution stages* (Figure 3, element 306).

Bhamidipati did not explicitly state *wherein each of the second predetermined number of instructions were received directly from one of the fetch stages.* Hayes demonstrated that it was known at the time of invention to directly couple a fetch stage to an execution stage (page 223-224, Figures 3.68-3.69) in a typical pipeline arrangement. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati's pipeline as a fetch stage directly coupled to an execution stage as found in Hayes' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement a standard (and thus well known/easy to implement) pipeline and Bhamidipati further indicated many different types pipeline stages can be used to implement the invention (column 3, lines 54-64).

6. Claims 2-6, 22-25, 28-30, 33-38, 41-43 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhamidipati et al. (USPN 6,112,295) in view of Hayes, John P., "Computer Architecture and Organization" and in further view of Kyker et al. (USPN 6,026,477).

In regard to claim 2, Bhamidipati and Hayes did not explicitly state *wherein said pipeline fetch circuit includes an instruction queue to store a predetermined maximum number of the instructions that are each ready to be processed by said pipeline fetch circuit.* Kyker demonstrated that it was known at the time of invention to provide a queue to temporarily store an instruction from memory before heading to later fetch stages (column 1, lines 14-26). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' pipeline with a prefetch queue as an intermediary between memory and the fetch stages as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to fetch as many instructions as possible to be waiting for processing at the earliest moment.

In regard to claim 3, Bhamidipati and Hayes disclosed the limitation *wherein an instruction can enter ... stage of processing independently of the movement of instructions through said pipeline execution circuit* (Bhamidipati: Figure 3, element 306). Bhamidipati and Hayes did not explicitly state *wherein said pipeline fetch circuit includes a pre-decode logic circuit to generate pre-decode signals for an instruction that is in a pre-decode stage of processing within said pipeline fetch circuit.* Kyker demonstrated that it was known at the time of invention to provide multiple decode stages (Figure 1, element 14). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' decoupling pipeline

with a pre-decode stage as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to construct pipelines with sufficient depth to quickly manage steps at the smallest unit-time level, including the decode process.

In regard to claim 4, Bhamidipati, Hayes and Kyker disclosed the limitations *wherein said pipeline fetch circuit includes a decode logic circuit coupled to said pre-decode logic circuit to generate decode signals for an instruction that is in a decode stage of processing within said pipeline fetch circuit* (Kyker: Figure 1, element 14), *and wherein an instruction can enter said decode stage of processing from said pre-decode stage of processing independently of the movement of instructions through said pipeline execution circuit* (Bhamidipati: Figure 3, element 306).

In regard to claim 5, Bhamidipati, Hayes and Kyker did not explicitly state the limitation *wherein said pipeline fetch circuit includes a first selection circuit coupled to said pre-decode logic circuit to allow an instruction to be received by said pre-decode logic circuit at a time determined by the system clock signal if said decode logic circuit is available to accept an instruction currently being executed by said pre-decode logic circuit.* Kyker demonstrated that it was known at the time of invention to for pipelines to stall (column 3, lines 40-45). A stall prevents a current stage from moving forward when the next stage is not available. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati, Hayes and Kyker's pipeline in such a

manner as a pre-decode stage can only process instructions if clock allows it and the next stage (decode stage) will accept the previous pre-decode processed instruction as suggested by Kyker's own teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated by the fact that stalls are common in pipeline technology.

In regard to claim 6, Bhamidipati, Hayes and Kyker did not explicitly state the limitation *wherein said pipeline fetch circuit includes a second selection circuit coupled to said decode logic circuit to allow an instruction to enter said decode stage of execution at a time determined by the system clock signal if said decode logic circuit is not processing another instruction*. Kyker demonstrated that it was known at the time of invention to for pipelines to stall (column 3, lines 40-45). A stall prevents a current stage from moving forward when the next stage is not available. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati, Hayes and Kyker's pipeline in such a manner as a stage can only process instructions if clock allows it and the next stage will accept the previous processed instruction as suggested by Kyker's own teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated by the fact that stalls are common in pipeline technology.

In regard to claim 22, Bhamidipati and Hayes disclosed the limitation *wherein one of the fetch stages includes instruction address generate logic* (Hayes: page 224, Figure 3.69;

page 223, section "Instruction pipelines" discusses instruction branching). Bhamidipati and Hayes did not explicitly state instruction address generate logic *to predict which instructions are to enter the fetch stages.* Kyker demonstrated that it was known at the time of invention to predict instructions to fetch (column 2, lines 1-8). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' instruction address generation logic as to predict instructions to fetch based upon branching as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to begin the initial phases of instruction processing on an instruction that will ultimately be executed and thus increase efficiency.

In regard to claim 23, Bhamidipati, Hayes and Kyker further disclosed the limitation *wherein the instruction address generate logic includes a circuit to clear ones of the fetch stages in response to a determination that instruction execution was re-directed* (Kyker: column 2, lines 54-66; Hayes: section "Instruction pipelines").

In regard to claim 24, Bhamidipati and Hayes disclosed the limitations:

- *including a memory to store instructions* (Hayes: page 223, second sentence in "Instruction pipelines");

Bhamidipati and Hayes did not explicitly state the limitations:

- *a queue coupled to the memory to temporarily store at least one instruction fetched from the memory; and*

- *a circuit coupled to the queue and to at least one of the fetch stages to fetch an instruction from the queue for presentation to at least one of the fetch stages.*

Kyker demonstrated that it was known at the time of invention to provide a queue to temporarily store an instruction from memory and a circuit coupled to the queue and a fetch stage to fetch instructions from the queue to the fetch stages (column 1, lines 14-26). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' pipeline with a prefetch queue as an intermediary between memory and the fetch stages as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to fetch as many instructions as possible to be waiting for processing at the earliest moment.

In regard to claim 25, Bhamidipati, Hayes and Kyker further disclosed the limitation *wherein the circuit coupled to the queue is capable of retrieving instructions from the queue for presentation to the at least one of the fetch stages regardless of whether instructions are advancing within the execution stages* (Bhamidipati's decoupling queue system Figure 3, element 306 allows the circuit to remain independent of the execution stages).

In regard to claim 28, Bhamidipati and Hayes disclosed the limitation *wherein one of the fetch stages includes address generate logic* (Hayes: page 224, Figure 3.69; page 223,

section "Instruction pipelines" discusses instruction branching). Bhamidipati and Hayes did not explicitly state instruction address generate logic *to predict which instructions are to enter the fetch stages.* Kyker demonstrated that it was known at the time of invention to predict instructions to fetch (column 2, lines 1-8). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' instruction address generation logic as to predict instructions to fetch based upon branching as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to begin the initial phases of instruction processing on an instruction that will ultimately be executed and thus increase efficiency.

In regard to claim 29, Bhamidipati, Hayes and Kyker further disclosed the limitation *wherein the address generate logic includes a circuit to flush one or more instructions from the fetch stages if it is determined that a misprediction occurred* (Kyker: column 2, lines 54-66; Hayes: section "Instruction pipelines").

In regard to claim 30, Bhamidipati and Hayes disclosed the limitations:

- *a memory* (Hayes: page 223, second sentence in "Instruction pipelines"); *and*

Bhamidipati and Hayes did not explicitly state:

- *a storage device coupled to one of the fetch stages and to the memory to store instructions retrieved from the memory, wherein a predetermined*

number of instructions may be stored within the storage device regardless of whether instructions are advancing within the fetch stages.

Kyker demonstrated that it was known at the time of invention to provide a queue to temporarily store an instruction from memory and a circuit coupled to the queue and a fetch stage to fetch instructions from the queue to the fetch stages (column 1, lines 14-26). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' pipeline with a prefetch queue as an intermediary between memory and the fetch stages as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to fetch as many instructions as possible to be waiting for processing at the earliest moment.

In regard to claim 33, Bhamidipati and Hayes disclosed the limitation *wherein one of the fetch stages includes instruction address generate logic* (Hayes: page 224, Figure 3.69; page 223, section "Instruction pipelines" discusses instruction branching). Bhamidipati and Hayes did not explicitly state instruction address generate logic *to determine which instructions are to enter the fetch circuit.* Kyker demonstrated that it was known at the time of invention to predict instructions to fetch (column 2, lines 1-8). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' instruction address generation logic as to predict instructions to fetch based upon branching as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to begin

the initial phases of instruction processing on an instruction that will ultimately be executed and thus increase efficiency.

In regard to claim 34, Bhamidipati, Hayes and Kyker disclosed the limitation *wherein the instruction address generate section includes a circuit to remove instructions from the fetch circuit during a pipeline flush operation* (Kyker: column 2, lines 54-64; Hayes: section "Instruction pipelines").

In regard to claim 35, Bhamidipati and Hayes disclosed the limitations:

- *a memory to store instructions* (Hayes: page 223, second sentence in "Instruction pipelines"); *and*

Bhamidipati and Hayes did not explicitly state:

- *a queue coupled to store instructions from the memory, the queue further being coupled to provide an instruction to the fetch circuit if one of the fetch stages is available and irrespective of whether an instruction is being provided from the fetch circuit to the execution circuit.*

Kyker demonstrated that it was known at the time of invention to provide a queue to temporarily store an instruction from memory and a circuit coupled to the queue and a fetch stage to fetch instructions from the queue to the fetch stages (column 1, lines 14-26). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' pipeline with a prefetch queue as an intermediary between memory and the fetch stages as found in Kyker's teaching. Thus, using

Bhamidipati's decoupling pipeline, the fetch circuit could continuously process instructions regardless of the execution stages provided the fetch circuit does not itself become saturated. This implementation would have been obvious because one of ordinary skill in the art would be motivated to fetch as many instructions as possible to be waiting for processing at the earliest moment.

In regard to claim 36, Bhamidipati, Hayes and Kyker disclosed *a circuit coupled to the queue to allow an instruction to be stored to the queue independently of whether an instruction is advancing within the fetch circuit* (Kyker: column 1, lines 14-26).

In regard to claim 37, Bhamidipati, Hayes and Kyker disclosed the limitation *wherein the circuit allows a predetermined maximum number of instructions to be stored to the queue independently of whether an instruction is advancing within the fetch circuit* (Kyker: column 1, lines 14-26).

In regard to claim 38, Bhamidipati, Hayes and Kyker did not explicitly state the limitation *wherein the one of the fetch stages includes a circuit to allow retrieval of an instruction from either the memory or from the queue.* Kyker demonstrated that it was known at the time of invention to flush a pipeline (column 2, lines 54-66). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati, Hayes and Kyker's pipeline with the ability to choose between fetching instructions directly from memory or from a prefetch queue as needed as suggested by

Kyker's teachings on pipeline flush. This implementation would have been obvious because one of ordinary skill in the art would be motivated to retrieve instructions from the queue if they existed and from memory directly if the queue had had to be emptied (it is fast to go to memory if the queue provides no use as a "middle-man").

In regard to claim 41, Bhamidipati and Hayes disclosed the limitations:

- *fetching an instruction from a memory* (Hayes: page 223, second sentence in "Instruction pipelines");

Bhamidipati and Hayes did not explicitly state:

- *storing the instruction within a queue; and*
- *retrieving the instruction from the queue to undergo a pre-execution operation within a predetermined one of the fetch stages*

Kyker demonstrated that it was known at the time of invention to provide a queue to temporarily store an instruction from memory and a circuit coupled to the queue and a fetch stage to fetch instructions from the queue to the fetch stages (column 1, lines 14-26). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' pipeline with a prefetch queue as an intermediary between memory and the fetch stages as found in Kyker's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to fetch as many instructions as possible to be waiting for processing at the earliest moment.

In regard to claim 42, Bhamidipati, Hayes and Kyker disclosed the limitation *wherein at least one of the storing and the retrieving step is performed independently of whether instructions are advancing between ones of the execution stages* (Kyker: column 1, lines 14-26; Bhamidipati: Figure 3, element 306).

In regard to claim 43, Bhamidipati, Hayes and Kyker disclosed the limitation *wherein ones of the steps are repeated for multiple instructions* (this is the nature of pipelines, to repeatedly perform steps for multiple instructions)

In regard to claim 45, Bhamidipati and Hayes did not explicitly state the limitations:

- *providing an indication that one or more predetermined operations are occurring within one or more of the execution stages; and*
- *in response to the indication, allowing instructions to advance within the fetch stages.*

Kyker demonstrated that it was known at the time of invention to for pipelines to stall (column 3, lines 40-45). A stall prevents a current stage from moving forward when the next stage is not available. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati, Hayes and Kyker's pipeline in such a manner as an indication being provided when the execution stages have/are stalled/busy as suggested by Kyker's own teaching. Clearly, Bhamidipati's pipeline allows some of the fetch stages to continue (Figure 3, element 306) depending on the placement of the decoupling queue. It would have been obvious to one of ordinary skill

in the art at the time of invention to provide a stall signal from the execution stages of Bhamidipati, Hayes and Kyker's pipeline and in response the fetch stages that are able to continue processing. This implementation would have been obvious because one of ordinary skill in the art would be motivated by the fact that stalls are common in pipeline technology and a decoupling queue is specifically designed to avoid stalls in the entire pipeline.

7. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bhamidipati et al. (USPN 6,112,295) in view of Hayes, John P., "Computer Architecture and Organization" in view of Kyker et al. (USPN 6,026,477) and in further view of Alferness et al. (USPN 5,577,259).

In regard to claim 7, Bhamidipati, Hayes and Kyker disclosed the limitation:

- *wherein said first selection circuit includes a control circuit to allow an instruction to enter said pre-decode stage of processing while ... ones of the instructions are not advancing to a next stage of execution within said pipeline execution circuit* (Figure 3, element 306).

Bhamidipati, Hayes and Kyker did not explicitly state the limitation

- *wherein said pipeline execution circuit includes a microcode-controlled sequencer to control execution of extended stages of execution of extended-mode ones of the instructions, wherein during said extended stages of execution, ones of the instructions being executed by said pipeline*

execution circuit are not advancing to a next stage of execution within said pipeline execution circuit

Alferness demonstrated that it was known at the time of invention to utilize taught a microcode-controlled sequencer (column 1, lines 24-31) and the sequencer to control extended stages of execution of extended-mode instructions (column 1, lines 26-31; extended cycle instructions are the extended-mode instructions) wherein during some stages of execution of the extended-mode instructions, instructions are not advancing within the execution circuit (column 4, lines 63-67). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati, Hayes and Kyker's decoupling pipeline with extended-mode instruction handling ability which might stop the progression of the execution stages as found in Alferness' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a processor, which included the flexibility and control of micro-code over extended type instructions (Alferness: column 2, lines 38-42; column 3, lines 35-41).

8. Claims 26, 31 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhamidipati et al. (USPN 6,112,295) in view of Hayes, John P., "Computer Architecture and Organization" and in further view of Alferness et al. (USPN 5,577,259).

In regard to claim 26, Bhamidipati and Hayes did not explicitly state the limitations:

- *wherein at least one of the execution stages includes a microcode-controlled sequencer to control execution of extended-mode instructions, and*
- *wherein during some stages of execution of the extended-mode instructions, instructions are not advancing within the execution circuit.*

Alferness demonstrated that it was known at the time of invention to utilize taught a microcode-controlled sequencer (column 1, lines 24-31) and the sequencer to control extended stages of execution of extended-mode instructions (column 1, lines 26-31; extended cycle instructions are the extended-mode instructions) wherein during some stages of execution of the extended-mode instructions, instructions are not advancing within the execution circuit (column 4, lines 63-67). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' decoupling pipeline with extended-mode instruction handling ability which might stop the progression of the execution stages as found in Alferness' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a processor, which included the flexibility and control of micro-code over extended type instructions (Alferness: column 2, lines 38-42; column 3, lines 35-41).

In regard to claim 31, Bhamidipati and Hayes did not explicitly state *wherein one of the execution stages includes a microcode sequencer to execute predetermined ones of the instructions in a manner that may temporarily affect movement of instructions within the execution stages.* Alferness demonstrated that it was known at the time of invention to

utilize taught a microcode-controlled sequencer (column 1, lines 24-31) and the sequencer to control extended stages of execution of extended-mode instructions (column 1, lines 26-31; extended cycle instructions are the extended-mode instructions) wherein during some stages of execution of the extended-mode instructions, instructions are not advancing within the execution circuit (column 4, lines 63-67). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' decoupling pipeline with extended-mode instruction handling ability which might stop the progression of the execution stages as found in Alferness' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a processor, which included the flexibility and control of micro-code over extended type instructions (Alferness: column 2, lines 38-42; column 3, lines 35-41).

In regard to claim 44, Bhamidipati and Hayes did not explicitly state *wherein one of the execution stages includes a microcode-controlled sequencer for executing extended-mode instructions, and further including executing one of the extended-mode instructions in a manner that temporarily delays the advancing of instructions between ones of the execution stages.* Alferness demonstrated that it was known at the time of invention to utilize taught a microcode-controlled sequencer (column 1, lines 24-31) and the sequencer to control extended stages of execution of extended-mode instructions (column 1, lines 26-31; extended cycle instructions are the extended-mode instructions) wherein during some stages of execution of the extended-mode instructions,

instructions are not advancing within the execution circuit (column 4, lines 63-67). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Bhamidipati and Hayes' decoupling pipeline with extended-mode instruction handling ability which might stop the progression of the execution stages as found in Alferness' teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to design a processor, which included the flexibility and control of micro-code over extended type instructions (Alferness: column 2, lines 38-42; column 3, lines 35-41).

Examiner's Response

9. The amendment filed 30 April 2003 is objected to under 35 U.S.C. 132 because it introduces new matter into the disclosure. 35 U.S.C. 132 states that no amendment shall introduce new matter into the disclosure of the invention. The added material which is not supported by the original disclosure is as follows: *instruction address generate logic* (claims 22, 23, 28, 29, 33 and 34); *circuit to allow retrieval of an instruction from either the memory or from the queue* (claim 38); and *indication that one or more operations are occurring within execution stages and in response to the indication, allowing instructions to advance within the fetch stages* (claim 45). Applicant is required to cancel the new matter in the reply to this Office Action.

10. Applicant's arguments with respect to claims 1-7 have been considered but are moot in view of the new ground(s) of rejection.

Corresp ndenc Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (703)305-3305. The examiner can normally be reached 7:30am - 5:00pm Monday thru Thursday and 7:30am - 4:00pm every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703)746-7239 for regular communications and (703)746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
June 13, 2003

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100